9. Railroad

Program Name: Railroad.java Input File: railroad.dat

You are in charge of constructing a high speed railroad across a desert. The desert is very flat, except for a number of rectangular plateaus that would impede the construction. The train goes very fast, so it must travel in a straight line the whole time. It would be very expensive to tunnel through the plateaus, so your boss wants to know how feasible it is to avoid them.

The train is 6 meters wide, and the engineers tell you it should have at least 2 meters of buffer on each side for safety. Thus, you have decided to break the desert up into a 10x10 meter grid. Your boss says you can either go across the desert vertically or horizontally, so long as there is a straight line of 10 x10 grid cells from one edge to the other that are not part of any plateau. How many ways can you make the train get across the desert?

The row number and column number start at zero. To cross the desert means either starting at the first row and going to the last row in a straight vertical path without going through a plateau or starting at the first column and going to the last column in a straight horizontal path without going through a plateau.

Input

The first line of input contains T, the number of test cases that follow.
The first line of each test case contains three integers N, M, and K. N and M are the number of rows and columns of the desert grid, and K is the number of plateaus in the desert.
The next K lines each contain a single plateau in 4 integers: R, C, W, L. R and C are the row and column of the minimum row/column corner of the plateau, and W and L are the width and length of the plateau. Thus, the plateau covers the intersection of rows R, R+1, R+2, …, R+W-1, and columns C, C+1, C+2, …, C+L-1.

Output

For each test case, print out the number of possible ways to construct a valid track for the train.

Constraints

```
1 <= T <= 10
1 <= N, M <= 10^5
0 <= K <= 50
0 <= R < N, 1 <= W <= N, 1 <= R + W <= N
0 <= C < M, 1 <= L <= M, 1 <= C + L <= M
```

Example Input File

```
4
1 1 1
0 0 1 1
2 1 0
3 3 2
0 0 1 1
2 2 1 1
3 3 2
0 0 1 1
2 1 1 2
```

Example Output to Screen

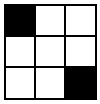
```
0
3
2
1
```

Explanation of Example Output

In the first case, the entire 1x1 desert was covered, so there is no valid path.

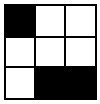
In the second case, the whole desert is open. Thus, you could go all the way down the 2 columns or 1 row.

In the third case, the desert looks like this:



So the only way to get across is down column 1 or across row 1.

In the fourth test case, the desert looks like this:



And thus, the only way to get to the other side is across row 1.